PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+"thread stack" local global root status

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before December 1998
Terms used **thread stack local global root status**

Found **44** of **93,559**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 44

Result page: **1** 2 3 next

Relevance scale

1 Sharing and protection in a single-address-space operating system
Jeffrey S. Chase, Henry M. Levy, Michael J. Feeley, Edward D. Lazowska
November 1994 **ACM Transactions on Computer Systems (TOCS)**, Volume 12 Issue 4
Full text available: pdf(2.87 MB) Additional Information: full citation, abstract, references, citings, index terms

This article explores memory sharing and protection support in Opal, a single-address-space operating system designed for wide-address (64-bit) architectures. Opal threads execute within protection domains in a single shared virtual address space. Sharing is simplified, because addresses are context independent. There is no loss of protection, because addressability and access are independent; the right to access a segment is determined by the protection domain in which a thread executes. T ...

Keywords: 64-bit architectures, capability-based systems, microkernel operating systems, object-oriented database systems, persistent storage, protection, single-address-space operating systems, wide-address architectures

2 Flexible register management for sequential programs
Donna J. Quammen, D. Richard Miller
April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture**, Volume 19 Issue 3
Full text available: pdf(968.52 KB) Additional Information: full citation, references, citings, index terms

3 A foundation for an efficient multi-threaded scheme system
Suresh Jagannathan, Jim Philbin
January 1992 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1992 ACM conference on LISP and functional programming**, Volume V Issue 1
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

We have built a parallel dialect of Scheme called STING that differs from its contemporaries in a number of important respects. STING is intended to be used as an operating system substrate for modern parallel programming languages. The basic concurrency management objects is STING are first-class lightweight threads of control and virtual processors (VPs). Unlike high-level concurrency structures, STING threads and VPs are not encumbered by complex synchronization protocols. ...

4 Implementation of Argus
B. Liskov, D. Curtis, P. Johnson, R. Scheifer
November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles**, Volume 21 Issue 5
Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

Argus is a programming language and system developed to support the construction and execution of distributed programs. This paper describes the implementation of Argus, with particular emphasis on the way we implement atomic actions, because this is where Argus differs most from other implemented systems. The paper also discusses the performance of Argus. The cost of actions is quite reasonable, indicating that action systems like Argus are practical.

5 The Flux OSKit: a substrate for kernel and language research
Bryan Ford, Godmar Back, Greg Benson, Jay Lepreau, Albert Lin, Olin Shivers
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5
Full text available: pdf(2.47 MB) Additional Information: full citation, references, citings, index terms

6 Implementing Ada 9X features using POSIX Threads: design issues
E. W. Giering, Frank Mueller, T. P. Baker
October 1993 **Proceedings of the conference on TRI-Ada '93**
Full text available: pdf(1.49 MB) Additional Information: full citation, references, citings, index terms

7 Supporting dynamic data structures on distributed-memory machines
Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren
March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 2
Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Compiling for distributed-memory machines has been a very active research area in recent years. Much of this work has concentrated on programs that use arrays as their primary data structures. To date, little work has been done to address the problem of supporting programs that use pointer-based dynamic data structures. The techniques developed for supporting SPMD execution of array-based programs rely on the fact that arrays are statically defined and directly addressable. Recursive data s ...

Keywords: dynamic data structures

8 Whole-program optimization for time and space efficient threads
Dirk Grunwald, Richard Neves
September 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 31 , 30 Issue 9 , 5
Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, citings, index terms

Modern languages and operating systems often encourage programmers to use *threads*, or independent control streams, to mask the overhead of some operations and simplify program structure. Multitasking operating systems use threads to mask communication latency, either with hardwares devices or users. Client-server applications typically use threads to simplify the complex control-flow that arises when multiple clients are used. Recently, the scientific computing community has started using ...

9 The shared-memory language pSather on a distributed-memory multiprocessor
Jerome A. Feldman, Chu-Cheow Lim, Thomas Rauber
January 1993 **ACM SIGPLAN Notices**, Volume 28 Issue 1
Full text available: pdf(349.93 KB) Additional Information: full citation, index terms

10 Space efficient conservative garbage collection
Hans-Juergen Boehm
June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation**, Volume 28 Issue 6
Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

We call a garbage collector conservative if it has only partial information about the location of pointers, and is thus forced to treat arbitrary bit patterns as though they might be pointers, in at least some cases. We show that some very inexpensive, but previously unused techniques can have dramatic impact on the effectiveness of conservative garbage collectors in reclaiming memory. Our most significant observation is that static data that appears to point to the heap should not result i ...

11 TeleNotes: managing lightweight interactions in the desktop
Steve Whittaker, Jerry Swanson, Jakov Kucan, Candy Sidner
June 1997 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 4 Issue 2
Full text available: pdf(1.01 MB) Additional Information: full citation, abstract, references, citings, index terms

Communication theories and technology have tended to focus on extended, formal meetings and have neglected a prevalent and vital form of workplace communication—namely, lightweight communication. Unlike formal, extended meetings, lightweight interaction is brief, informal, unplanned, and intermittent. We analyze naturalistic data from a study of work-place communication and derive five design criteria for lightweight interaction systems. These criteria require that systems for lightwe ...

Keywords: audio, awareness, computer-media spaces, conversation management, impromptu communication, informal communication, interpersonal communications, lightweight interaction, mediated communication, remote collaboration, task management, video

12 A memory-efficient real-time non-copying garbage collector
Tian F. Lim, Przemysław Pardyak, Brian N. Bershad
October 1998 **ACM SIGPLAN Notices , Proceedings of the 1st international symposium on Memory management**, Volume 34 Issue 3
Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

Garbage collectors used in embedded systems such as Personal Java and Inferno or in operating systems such as SPIN must operate with limited resources and minimize their impact on application performance. Consequently, they must maintain short real-time pauses, low overhead, and a small memory footprint. Most garbage collectors, including the

Treadmill algorithm, are inadequate because they sacrifice space for time. We have implemented a new Treadmill variant that provides good memory utilizatio ...

Keywords: garbage collection, operating systems, real-time, treadmill

13 Pthreads for dynamic and irregular parallelism
Girija J. Narlikar, Guy E. Blelloch
November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: html(82.60 KB) Additional Information: full citation, abstract, references, citings

High performance applications on shared memory machines have typically been written in a coarse grained style, with one heavyweight thread per processor. In comparison, programming with a large number of lightweight, parallel threads has several advantages, including simpler coding for programs with irregular and dynamic parallelism, and better adaptability to a changing number of processors. The programmer can express a new thread to execute each individual parallel task; the implementation dyn ...

Keywords: Pthreads, dynamic scheduling, irregular parallelism, lightweight threads, multithreading, space efficiency

14 KDB: a multi-threaded debugger for multi-threaded applications
Peter A. Buhr, Martin Karsten, Jun Shih
January 1996 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**
Full text available: pdf(991.10 KB) Additional Information: full citation, references, citings, index terms

15 The effectiveness of affinity-based scheduling in multiprocessor network protocol processing (extended version)
James D. Salehi, James F. Kurose, Don Towsley
August 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 4
Full text available: pdf(1.71 MB) Additional Information: full citation, references, citings, index terms

16 The Amber system: parallel programming on a network of multiprocessors
J. Chase, F. Amador, E. Lazowska, H. Levy, R. Littlefield
November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles**, Volume 23 Issue 5
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a programming system called Amber that permits a single application program to use a homogeneous network of computers in a uniform way, making the network appear to the application as an integrated multiprocessor. Amber is specifically designed for high performance in the case where each node in the network is a shared-memory multiprocessor. Amber shows that support for loosely-coupled multiprocessing can be efficiently realized using an obje ...

17 Efficient large-scale process-oriented parallel simulations
Kalyan S. Perumalla, Richard M. Fujimoto
December 1998 **Proceedings of the 30th conference on Winter simulation**

Full text available: pdf(63.76 KB) Additional Information: full citation, references, citings, index terms

18 Real-time concurrent collection on stock multiprocessors
A. W. Appel, J. R. Ellis, K. Li
June 1988 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation**, Volume 23 Issue 7
Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings, index terms

We've designed and implemented a copying garbage-collection algorithm that is efficient, real-time, concurrent, runs on commercial uniprocessors and shared-memory multiprocessors, and requires no change to compilers. The algorithm uses standard virtual-memory hardware to detect references to "from space" objects and to synchronize the collector and mutator threads. We've implemented and measured a prototype running on SRC's 5-processor Firefly. It will be straightforward to merg ...

19 Object and native code thread mobility among heterogeneous computers (includes sources)
B. Steensgaard, E. Jul
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5
Full text available: pdf(1.50 MB) Additional Information: full citation, references, citings, index terms

20 Application downloading
Robert Balzer, Alvin Cooperband, Martin Feather, Philip London, David Wile
March 1981 **Proceedings of the 5th international conference on Software engineering**
Full text available: pdf(774.21 KB) Additional Information: full citation, abstract, references, index terms

The purpose of our research is to investigate the feasibility of this methodology for distributed application development by examining possible approaches to carrying out such developments. In particular, our emphasis lies in transformation technology, wherein user-invoked source-to-source transformations are applied to the partitioned system for the purpose of optimizing the transactions required to effect the distributed behavior. In this way, the application downloading system need only ...

Results 1 - 20 of 44 Result page: **1** 2 3 next

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+"thread stack" local global root status

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before December 1998
Terms used **thread stack** **local** **global** **root** **status**

Found **44** of **93,559**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 21 - 40 of 44

Result page: previous 1 **2** 3 next

Relevance scale

21 A dataflow language with object-based extension and its implementation on a commercially available parallel machine
Shigeru Kusakabe, Taku Nagai, Yoshihiro Yamashita, Rin-ichiro Taniguchi, Makoto Amamiya
July 1995 **Proceedings of the 9th international conference on Supercomputing**
Full text available: pdf(979.70 KB) Additional Information: full citation, references, index terms

22 Mostly parallel garbage collection
Hans-J. Boehm, Alan J. Demers, Scott Shenker
May 1991 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1991 conference on Programming language design and implementation**, Volume 26 Issue 6
Full text available: pdf(862.08 KB) Additional Information: full citation, references, citings, index terms

23 Supporting thousands of threads using a hybrid stack sharing scheme
Kam-Fai Wong, Benoît Dageville
April 1994 **Proceedings of the 1994 ACM symposium on Applied computing**
Full text available: pdf(735.97 KB) Additional Information: full citation, references, index terms

24 OpenMP on networks of workstations
Honghui Lu, Y. Charlie Hu, Willy Zwaenepoel
November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(202.91 KB) Additional Information: full citation, abstract, references, citings

We describe an implementation of a sizable subset of OpenMP on networks of workstations (NOWs). By extending the availability of OpenMP to NOWs, we overcome one of its primary drawbacks compared to MPI, namely lack of portability to environments other than hardware shared memory machines. In order to support OpenMP execution on NOWs, our compiler targets a software distributed shared memory system (DSM) which provides multi-threaded execution and memory consistency.This paper presents two contri ...

25

Checkpoint and recovery methods in the ParaSol simulation system
Edward Mascarenhas, Felipe Knop, Reuben Pasquini, Vernon Rego
December 1997 **Proceedings of the 29th conference on Winter simulation**
Full text available: pdf(851.44 KB) Additional Information: full citation, references, citings, index terms

26 Compiling nested data-parallel programs for shared-memory multiprocessors
Siddhartha Chatterjee
July 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 3
Full text available: pdf(4.17 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: compilers, data parallelism, shared-memory multiprocessors

27 Dynamic node reconfiguration in a parallel-distributed environment
Michael J. Feeley, Brian N. Bershad, Jeffrey S. Chase, Henry M. Levy
April 1991 **ACM SIGPLAN Notices , Proceedings of the third ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 26 Issue 7
Full text available: pdf(901.48 KB) Additional Information: full citation, references, citings, index terms

28 Using high performance GIS software to visualize data: a hands-on software demonstration
Linda Burton, William Hatchett, Mari Hobkirk, Charles Powell
November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: html(80.49 KB) Additional Information: full citation, abstract, references

Since 1995 Wheat Ridge High School (WRHS) students have participated in a mapping project involving local open space, in conjunction with NASA. Students have learned to use *Idrisi,* a Geographical Imaging Systems (GIS) software, as well as other GIS programs *Arc View* and *Multispec,* to plan the location of a trail along Colorado's front range.As this project has progressed, students have learned the GIS technology as well as many science issues related to trail mapping. Simila ...

29 Efficient process interaction with threads in parallel discrete event simulation
Reuben Passqini, Vernon Rego
December 1998 **Proceedings of the 30th conference on Winter simulation**
Full text available: pdf(80.35 KB) Additional Information: full citation, references, index terms

30 Optimistic active messages: a mechanism for scheduling communication with computation
Deborah A. Wallach, Wilson C. Hsieh, Kirk L. Johnson, M. Frans Kaashoek, William E. Weihl
August 1995 **ACM SIGPLAN Notices , Proceedings of the fifth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 30 Issue 8
Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, citings, index terms

Low-overhead message passing is critical to the performance of many applications. Active

Messages reduce the software overhead for message handling: messages are run as handlers instead of as threads, which avoids the overhead of thread management and the unnecessary data copying of other communication models. Scheduling the execution of Active Messages is typically done by disabling and enabling interrupts, or by polling the network. This primitive scheduling control, combined with the fac ...

31 Clarity MCode: a retargetable intermediate representation for compilation
Brian T. Lewis, L. Peter Deutsch, Theodore C. Goldstein
March 1995 **ACM SIGPLAN Notices , Papers from the 1995 ACM SIGPLAN workshop on Intermediate representations**, Volume 30 Issue 3
Full text available: pdf(948.64 KB) Additional Information: full citation, citings, index terms

32 Improving prediction for procedure returns with return-address-stack repair mechanisms
Kevin Skadron, Pritpal S. Ahuja, Margaret Martonosi, Douglas W. Clark
November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.66 MB) Additional Information: full citation, references, citings, index terms

33 User-level threads on a general hardware interface
K. R. Mayes, S. Quick, B. C. Warboys
October 1995 **ACM SIGOPS Operating Systems Review**, Volume 29 Issue 4
Full text available: pdf(539.99 KB) Additional Information: full citation, abstract, index terms

Moving resource management out of the operating system kernel facilitates a high degree of customisation. The lowest layer of the Arena system provides an abstract interface to conventional processor hardware (Mayes, 1993; Quick, 1995). The idea is to encapsulate the hardware behind an interface with certain low-level concepts which are generally applicable to any processor. Localization of hardware-dependency has the effect of increasing modularity and thus portability. This encapsulation, term ...

34 Better operating system features for faster network servers
Gaurav Bangs, Peter Druschel, Jeffrey C. Mogul
December 1998 **ACM SIGMETRICS Performance Evaluation Review**, Volume 26 Issue 3
Full text available: pdf(911.11 KB) Additional Information: full citation, abstract, citings, index terms

Widely-used operating systems provide inadequate support for large-scale Internet server applications. Their algorithms and interfaces fail to efficiently support either event-driven or multi-threaded servers. They provide poor control over the scheduling and management of machine resources, making it difficult to provide robust and controlled service. We propose new UNIX interfaces to improve scalability, and to provide fine-grained scheduling and resource management.

35 Interrupts as threads
Steve Kleiman, Joe Eykholt
April 1995 **ACM SIGOPS Operating Systems Review**, Volume 29 Issue 2
Full text available: pdf(492.24 KB) Additional Information: full citation, abstract, citings, index terms

Most operating system implementations contain two fundamental forms of asynchrony; processes (or equivalently, internal threads) and interrupts. Processes (or threads) synchronize using primitives such as mutexes and condition variables, while interrupts are synchronized by preventing their occurrence for a period of time. The latter technique not only is expensive, but it locks out interrupts on the possibility that an interrupt will occur and interfere with the particular critical section of c ...

36 Miscellaneous: Language- and application-oriented resource management for parallel architectures □
Ken Mayes, Stuart Quick, James Bridgland, Andy Nisbet
September 1994 **Proceedings of the 6th workshop on ACM SIGOPS European workshop: Matching operating systems to application needs**
Full text available: pdf(544.35 KB) Additional Information: full citation, abstract, references

General-purpose operating systems, such as UNIX, which evolved on single processor machines, have made the transition, in one form or another, to parallel architectures (e.g. Rothnie, 1992; Holman, 1992). However, it is not clear that all users of parallel architectures require the virtual machine presented by a general-purpose operating system (e.g. Bryant et al., 1991). It is unfortunate if such users are given the alternatives of either compromising with whatever operating system interface is ...

37 A performance evaluation of lock-free synchronization protocols □
Anthony LaMarca
August 1994 **Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing**
Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

38 The performance of an object-oriented threads package □
John E. Faust, Henry M. Levy
September 1990 **ACM SIGPLAN Notices , Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications**, Volume 25 Issue 10
Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

Presto is an object-oriented threads package for writing parallel programs on a shared-memory multiprocessor. The system adds thread objects and synchronization objects to C++ to allow programmers to create and control parallelism. Presto's object-oriented structure, along with its user-level thread implementation, simplifies customization of thread management primitives to meet application-specific needs. The performance of thread primitives is crucial for parallel programs with ...

39 FTL: a multithreaded environment for parallel computation □
Ivan Kalas, Eshrat Arjomandi, Guang R. Gao, Bill O'Farrell
October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(233.30 KB) Additional Information: full citation, abstract, references, citings, index terms

The arrival of high-performance "killer micros" and the availability of high-performance networks (e.g., ATM) offer potential for building clusters of workstations with a significantly higher level of scalability than before. A promising approach to exploiting parallel computation on these systems is to use multithreading to overlap computation and communication while offering a simple programming model that smoothly integrates these two functions.This paper describes the design and implementati ...

40 Scheduling for cache affinity in parallelized communication protocols □
James D. Salehi, James F. Kurose, Don Towsley
May 1995 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1995**

ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, Volume 23 Issue 1

Full text available: pdf(242.17 KB) Additional Information: full citation, abstract, references, citings, index terms

We explore processor-cache affinity scheduling of parallel network protocol processing in a setting in which protocol processing executes on a shared-memory multiprocessor concurrently with a general workload of non-protocol activity. We find that affinity scheduling can significantly reduce the communication delay associated with protocol processing, enabling the host to support a greater number of concurrent streams and to provide a higher maximum throughput to individual streams. In addition, ...

Results 21 - 40 of 44 Result page: previous 1 **2** 3 next

PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+"thread stack" local global root status

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before December 1998
Terms used **thread stack local global root status**

Found **44** of **93,559**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 41 - 44 of 44

Result page: previous 1 2 **3**

Relevance scale □□□□□

41 A minimal TTL processor for architecture exploration □

Bradford J. Rodriguez

April 1994 **Proceedings of the 1994 ACM symposium on Applied computing**

Full text available: pdf(329.43 KB) Additional Information: full citation, references, index terms

Keywords: CPU, architecture, education, processors

42 Another C Threads library □

G. Cattaneo, G. Di Giore, M. Ruotolo

December 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 12

Full text available: pdf(674.39 KB) Additional Information: full citation, abstract, index terms

In this paper we present the main results achieved with the implementation of an efficient multithread library, hereinafter called DIA_Thread library, built on the top of a shared memory symmetric multiprocessor environment. The idea which led us to this project was the lack of a *minimal* common layer which, as far the multitasking concerns, should help us to make our applications portable among different architectures and/or operating systems. In fact, there are more and more multiprocess ...

43 Using continuations to implement thread management and communication in operating systems □

Richard P. Draves, Brian N. Bershad, Richard F. Rashid, Randall W. Dean

September 1991 **ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles**, Volume 25 Issue 5

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings, index terms

We have improved the performance of the Mach 3.0 operating system by redesigning its internal thread and interprocess communication facilities to use *continuations* as the basis for control transfer. Compared to previous versions of Mach 3.0, our new system consumes 85% less space per thread. Cross-address space remote procedure calls execute 14% faster. Exception handling runs over 60% faster.In addition to improving system performance, we have used continuations to generalize many contro ...

44 Ethercom: a study of audio processes and synchronization □
Richard Rybacki, Kay A. Robbins, Steven Robbins
March 1993 **ACM SIGCSE Bulletin , Proceedings of the twenty-fourth SIGCSE technical symposium on Computer science education**, Volume 25 Issue 1
Full text available: pdf(498.96 KB) Additional Information: full citation, references, index terms

Results 41 - 44 of 44 Result page: previous 1 2 **3**

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark Office

IEEE Xplore®
1 Million Documents
1 Million Users

» Search Results

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore®

○- Home
○- What Can I Access?
○- Log-out

Tables of Contents

○- Journals & Magazines
○- Conference Proceedings
○- Standards

Search

○- By Author
○- Basic
○- Advanced
○- CrossRef

Member Services

○- Join IEEE
○- Establish IEEE Web Account
○- Access the IEEE Member Digital Library

IEEE Enterprise

○- Access the IEEE Enterprise File Cabinet

Your search matched **2** of **1131693** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

thread <and> stack <and> heap Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Thread migration in the presence of pointers**
Cronk, D.; Haines, M.; Mehrotra, P.;
System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on , Volume: 1 , 7-10 Jan. 1997
Pages:292 - 298 vol.1

[Abstract] [PDF Full-Text (808 KB)] **IEEE CNF**

2 **A Hybrid Execution Model for Fine-Grained Languages on Distributed Memory Multicomputers**
Plevyak, J.; Karamcheti, V.; Xingbin Zhang; Chien, A.A.;
Supercomputing, 1995. Proceedings of the IEEE/ACM SC95 Conference , 03-06 Dec. 1995
Pages:41 - 41

[Abstract] [PDF Full-Text (808 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Welcome to IEEE Xplore®
- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents
- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search
- ○ By Author
- ○ Basic
- ○ Advanced
- ○ CrossRef

Member Services
- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

IEEE Enterprise
- ○ Access the IEEE Enterprise File Cabinet

Your search matched **2** of **1131693** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

thread <and> stack <and> local **Search**

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **HRTEM study of strain energy distribution on the self-organized In-rich quantum dots of the InGaN/GaN quantum wells**
Yen-Sheng Lin; Kung-Jen Ma; Chih-Chung Yang; Weirich, T.E.;
Numerical Simulation of Optoelectronic Devices, 2004. NUSOD '04. Proceedings of the 4th International Conference on , 24-26 Aug. 2004
Pages:88 - 89

[Abstract] [PDF Full-Text (290 KB)] **IEEE CNF**

2 **Building distributed embedded systems with RTLinux-GPL**
Perez, S.; Vila, J.;
Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference , Volume: 1 , 16-19 Sept. 2003
Pages:161 - 168 vol.1

[Abstract] [PDF Full-Text (619 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark Office

IEEE Xplore®
1 Million Documents
1 Million Users

» Search Results

Help FAQ Terms IEEE Peer Review Quick Links

Welcome to IEEE Xplore

- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents

- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search

- ○ By Author
- ○ Basic
- ○ Advanced
- ○ CrossRef

Member Services

- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

IEEE Enterprise

- ○ Access the IEEE Enterprise File Cabinet

Your search matched **1** of **1131693** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

thread <and> stack <and> global [Search]

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **The Augmint multiprocessor simulation toolkit for Intel x86 architectures**
Nguyen, A.-T.; Michael, M.; Sharma, A.; Torrellas, J.;
Computer Design: VLSI in Computers and Processors, 1996. ICCD '96. Proceedings., 1996 IEEE International Conference on , 7-9 Oct. 1996
Pages:486 - 490

[Abstract] [PDF Full-Text (564 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Google **Web** Images Groups News Froogle Local^New!^ more »

"thread stack" "thread heap" local global sta Search Advanced Search Preferences

Web Results **1** - **6** of **6** for "**thread stack**" "**thread heap**" **local global status**. (**0.56** seconds)

Tip: Try removing quotes from your search to get more results.

[PDF] Developing Platform Consistent Multithreaded Applications: Memory ...
File Format: PDF/Adobe Acrobat - View as HTML
... These functions maintain a per-**thread heap** attached to each ... management of this **global** storage is the programmer's responsibility. ...
cache-www.intel.com/cd/00/00/ 05/15/51533_chapter_5_memory_management02.pdf - Similar pages

[PS] Computer Systems Laboratory Cornell University, Ithaca, NY 14853 ...
File Format: Adobe PostScript - View as Text
... prefix to all **global** and static-**local** variable declarations. ... Additionally, similar to a **thread's stack** region, an individual **thread's heap** may ...
www.csl.cornell.edu/TR/CSL-TR-2001-1016.ps - Similar pages

[PS] Adding Threads to Standard ML Eric C. Cooper J. Gregory Morrisett ...
File Format: Adobe PostScript - View as Text
... aligned in such a way that by masking a **thread's stack** pointer appropriately, ... In addition, each **thread's heap** can have pointers into other threads' ...
www.eecs.harvard.edu/~greg/papers/jgmorris-mlthreads.ps - Similar pages

[PS] Region-Based Memory Management for Real-Time Java by William S ...
File Format: Adobe PostScript - View as Text
... the scoped memory must be present in the **thread's stack** of memory areas. ... With Without With **Thread Heap** Checks Heap Checks All Checks Benchmark Type ...
flexc.lcs.mit.edu/Harpoon/papers/wes-thesis.ps - Similar pages

[PS] Resource Control of Untrusted Code in an Open Programmable Network ...
File Format: Adobe PostScript - View as Text
... structure is going to be allocated in association with a **thread stack**; ... popping the heap record at the top of the current **thread's heap** stack and ...
www.paulmenage.org/papers/thesis.ps - Similar pages

kernel.h File Reference
... enough memory from the current **thread's heap** for nmemb ... should include space for the **thread stack** and **local** ... was previously allocated on the thread **local** heap by ...
www.cloudcaptech.com/Datasheets/uBurst/kernel_8h.html - 94k - Supplemental Result - Cached - Similar pages

"thread stack" "thread heap" loca Search